

1     **REMARKS**

2             The Office Action of November 22, 2005, has been carefully considered.  
3     Claims 1-23 are pending in the application.

4             Claims 1-23 were rejected under 35 U.S.C. §103(a) as being unpatentable  
5     over U.S. Patent No. 6,314,460 to Knight et al. (hereinafter referred to as the  
6     Knight reference) in view of US Patent Application No. 2005/0060693 A1 to  
7     Robison et al. (hereinafter referred to as the Robison reference).

8             The Applicant submits the following amendments and remarks to traverse  
9     the above rejections. The Applicant respectfully requests reconsideration and  
10    allowance of the subject application. This Amendment is believed to be fully  
11    responsive to all issues raised in the Office Action dated November 22, 2005.

12  
13    **Claim Rejections Under 35 USC §103(a)**

14            Claims 1-23 were rejected under 35 U.S.C. §103(a) as being  
15    unpatentable over the Knight reference in view of the Robison reference. For the  
16    reasons that follow, the Applicant respectfully disagrees that the subject matter of  
17    the above claims is obvious given the above cited references.

18  
19    **Rejection of Independent Claim 1 and its Dependent Claims 2-13**

20            The Examiner contends that the Knight reference teaches each of the  
21    elements recited in Claim 1, except that it does not teach “receiving a command  
22    string” or “separating the command string into one or more string components”.  
23    The Examiner contends that the Robison reference teaches “receiving a command  
24    string” and “separating the command string into one or more string components”.  
25    Then, the Examiner contends that it would have been obvious to a person having

1 ordinary skill in the art at the time the invention was made to have modified the  
2 Knight reference by the teaching of the Robison reference because do so would  
3 **enable the system to be distributed among remote resources, wherein**  
4 **command (strings) are generated by various entities of the system, broken**  
5 **down (separated) into various components, and are parsed (transmitted and**  
6 **received) by the resolving component”** (emphasis added). In addition, the  
7 Examiner contends that the Robison reference teaches that separating the  
8 parameter parsing logic from the actual action handler logic leads to better  
9 separation of concerns and OO-designs.

10 In order to best describe why the Applicant disagrees with the Examiner’s  
11 contention, the Applicant again briefly describes an overview of the invention and  
12 then describes certain aspects in further detail. The Applicant then contrasts the  
13 claimed invention with the Knight and the Robison references that were cited by  
14 the Examiner in rejecting Claims 1-13.

15 In overview, the present invention is directed at an extended type manager  
16 that is configured to access precisely parse-able input and to correlate the precisely  
17 parse-able input with a requested data type. Page 19, lines 3-7. The extended type  
18 manager may perform this function in response to a request from a parser, a script  
19 engine, or a pipeline processor. Page 60, lines 12-14. When a pipeline processor  
20 requests this functionality, the extended type manager resolves partially  
21 unresolved objects that are piped through the pipeline of the operating  
22 environment from one object-based command to the next object-based command.  
23 In addition, strings specified via the object-based command pipeline, may affect  
24 the processing of incoming objects. For example, property paths specify  
25 additional processing that is performed on a direct property of an incoming object

1 that was originally generated by a prior command. Page 61 line 20 to Page 62,  
2 line 1. The extended type manager also allows new data types to be incorporated  
3 into the operating system by various external sources. Page 20, lines 16-19. Each  
4 external source may register their unique structure within a type metadata and  
5 provide code. When the object is queried, the extended type manager reviews the  
6 type metadata to determine whether the object has been registered. If the object is  
7 not registered within the type metadata, reflection is performed. Otherwise,  
8 extended reflection is performed. Page 20, line 23 to Page 21, line 2. Thus,  
9 depending on the input type, the type metadata describes how the extended type  
10 manager should query various types of precisely parse-able input to obtain the  
11 desired properties for creating an object. Page 21, lines 6-9. In addition to  
12 providing extended types, the extended type manager provides additional query  
13 mechanisms, such as a **property path mechanism**, a **key mechanism**, a **compare**  
14 **mechanism**, a **conversion mechanism**, a **globber mechanism**, a **relationship**  
15 **mechanism**, and a **property set mechanism**. Page 21, lines 13-16.

16 In contrast, the Knight reference is directed at an analyzer for a storage  
17 network attached to a host computer system through multiple controllers that  
18 receives information from each controller concerning a shared storage network  
19 bus, and resolves incomplete information received from one controller using  
20 information received by another controller, as described in the Abstract. The  
21 “**storage network as used herein is an interconnected group of storage devices**  
22 **and controllers**”, as described in Column 5, lines 40-41. With this configuration,  
23 “**it is possible for a host to communicate with any storage device in a storage**  
24 **network to which the host is connected, without crossing another host’s**  
25 **backplane bus.**” While the Knight reference acknowledges that the network

1 configurations that are shown are merely illustrative, it states that the “number of  
2 host systems, I/O controllers, buses, and storage devices may vary considerably”.  
3 Thus, the shared network may be various configurations of host systems, I/O  
4 controllers, busses, and storage devices.

5 Again, the Applicant disagrees with the Examiner’s correlation of “the  
6 shared storage network” with the “interactive operating environment” recited in  
7 Claim 1. To further clarify, the Applicant has amended Claim 1 to recite  
8 “receiving a set of objects output from a prior command via an object-based  
9 command pipeline” and “processing the set of objects using an operating  
10 environment mechanism to resolve each object into a data type”. In other words,  
11 objects output from a prior command are received and are processed using an  
12 operating environment mechanism. Claim 11 further clarifies that the set of  
13 objects is received as input to a subsequent command in the object-based  
14 command pipeline after processing the set of objects using the operating  
15 environment mechanism. Thus, the claimed invention is not directed at parsing  
16 the parameters entered on a command line into objects.

17 Upon review of the Robison reference, the Applicant contends that the  
18 Robison reference merely discloses the parsing of parameters of a command string  
19 into objects. For example, paragraph [0018] states the following:

20  
21 [A]n embodiment of the present invention provides a command-line  
22 (command string) processing system for an OO environment. A  
23 command processor receives a command-string that is parsed into  
24 character string tokens. A parameter-handler (a type of parser) then  
25 attempts to match each successive token against command syntax  
descriptions that are loaded from syntax files. If the first token is  
matched against the first item of a command, whether that item is  
defined to be a command keyword or a parameter, then the  
parameter-handler tries to match the next token against the next item

1 in the command. This iterative matching process continues until no  
2 more matching can be performed. If all specified tokens have been  
3 matched successfully against the command syntax, then it is thus  
4 determined that the syntax is indeed that of the specified command.  
5 But if no match is found for one of the tokens, then the command  
6 processor continues its attempts to match the command-string with  
7 other syntax descriptions. If the entered command-string does not  
8 match any syntax description, then an error is indicated and a help  
9 message, e.g., proper usage of the attempted command or the  
10 command that most closely matched an invalid command, is  
11 outputted to the user or external calling module.

12  
13 The Robison reference further explains that “the specified token should  
14 exactly match the keyword expected by the command processor at that position  
15 and/or context within the command string” in paragraph [0019]. In paragraph  
16 [0021], the Robison describes the goal of their command string parsing as follows:

17  
18 Such a command-line processing system can successfully process  
19 new command-strings when syntax descriptions for such new  
20 commands are entered in the syntax files. New commands are those  
21 that were previously unsupported by the command-line processing  
22 system. The parameter-handling modules can be leveraged and  
23 reused by syntax descriptions. This can promote object-oriented  
24 design goals and substantially separate command-string parsing and  
25 processing concerns from the actual code for the command  
execution. The command execution code receives a set of data  
objects on which it can operate rather than a set of tokens that  
it must itself validate and convert to data objects.

26  
27 The Applicant contends that neither the Knight reference nor the Robison  
28 reference nor any permissible combination of both, teach “receiving a set of  
29 **objects output from a prior command via an object-based command pipeline**”  
30 and “**processing the set of objects using an operating environment mechanism**”

1 **to resolve each object in the set into a data type**” (emphasis added) as recited in  
2 Claim 1. For example, the Robison reference discloses command string parsing  
3 that separates parameter parsing logic from the actual action handler logic. This  
4 command string parsing disclosed in the Robison reference does not teach or  
5 suggest the limitations recited in Claim 1. For example, there is no mention of  
6 outputting a set of objects by a prior command. Rather, the objects discussed in  
7 Robison are objects created from the parameters entered on the command line. In  
8 addition, there is no mention of a “pipeline”, let alone an object-based command  
9 “pipeline” as recited in Claim 1.

10 In summary, the Examiner has not cited any reference that teaches or  
11 suggests the claimed invention. In fact, even if all of the cited references could be  
12 combined, their combined teachings could not possibly suggest the present  
13 invention. In addition, there is no suggestion or motivation to combine these  
14 references. Thus, for any of the reasons above, the Applicant contends that the  
15 Knight reference, whether considered alone or with any permissible combination  
16 of prior art of record, including the Robison reference, does not teach or suggest  
17 each limitation recited in independent Claim 1. Therefore, the Applicant  
18 respectfully submits that the §103 rejection of independent Claim 1 is improper,  
19 and respectfully requests reconsideration and withdrawal of this rejection.

20 Furthermore, the dependent Claims 2-13 of Claim 1 include other limitations  
21 that are not taught or suggested by the prior art of record. For example, Claims 5-8  
22 recite “receiving a string via the object-based command pipeline”. In contrast with  
23 the teachings in the Robison reference, this string is not parsed into objects, but  
24 rather the string affects the processing of the set of objects output from a prior  
25 command via the object-based command pipeline. Claim 5 recites “the string

1 includes a wildcard” and “processing by the mechanism comprises producing a  
2 subset of the set of object”. Claim 6 recites “the string includes a property set”  
3 and “processing by the mechanism comprises identifying a plurality of properties  
4 associated with the property set and processing the set of objects based on the  
5 plurality of properties.” Claim 7 recites “the string includes a relation” and  
6 “processing by the mechanism comprises finding items that the set of objects  
7 consume based on the relation.” Claim 8 recites “the string comprises a property  
8 path, the property path comprises a series of components that provide navigation  
9 to a desired property of each object in the set”. Thus, the string is not parsed into  
10 an object as disclosed by the Robison reference. Rather, *the string affects the*  
11 *processing of the set of objects output from a prior command*. Claim 11 further  
12 recites that the “set of objects is received as input to a subsequent command in the  
13 object-based command pipeline after processing the set of objects using the  
14 operating environment mechanism”. The Robison reference does not disclose, a  
15 pipeline, an object-based command pipeline, a set of object received as input to a  
16 subsequent command, and processing the set of object using the operating  
17 environment mechanism.

18 Therefore, for at least the above reasons, Applicant respectfully submits that  
19 the §103 rejections of dependent Claims 2-13 is improper, and respectfully requests  
20 reconsideration and withdrawal of this rejection.  
21  
22  
23  
24  
25

## Rejection of Independent Claims 14 and 19 and their Dependent Claims

The Examiner contends that the Knight reference teaches each of the elements recited in independent Claims 14 and 19, except that it does not teach “receiving parseable input”. The Examiner contends that the Robison reference teaches “receiving parseable input”. Then, the Examiner again contends that it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified the Knight reference by the teaching of the Robison reference because including receiving parseable input, would **enable the system to be distributed among remote resources, wherein input command (strings) are generated by various entities of the system and parsed (transmitted and received) by the resolving component**”. (emphasis added)

The Examiner now contends that the shared storage networks teaches an operating environment as recited in Claims 14 and 19. Without out unnecessarily repeating the above arguments for independent Claim 1, the Applicant states that the applicable arguments above also apply to these claims.

The Applicant has amended independent Claims 14 and 19 to clarify that the parseable input is received as “output from a prior command via an object-based command pipeline within an operating environment”. As discussed above, the Robison reference discloses a command string parsing method where strings entered on a command line are converted to objects before execution by a command. This does not teach or suggest the recited “object-based command pipeline” or “receiving parseable input output from a prior command” as recited in Claims 14 and 19.

In summary, the Examiner has not cited any reference that teaches or suggests the claimed invention. In fact, even if all of these references could be



1 combined, their teachings could not possibly suggest the present invention. In  
2 addition, there is no suggestion or motivation to combine these references. Thus,  
3 for at least any of the above reasons, the Applicant contends that the Knight  
4 reference, whether considered alone or with any permissible combination of prior  
5 art of record, including the Robison reference, does not teach or suggest each  
6 limitation recited in independent Claims 14 and 19. Therefore, the Applicant  
7 respectfully submits that the §103 rejection of independent Claims 14 and 19 is  
8 improper, and respectfully requests reconsideration and withdrawal of this  
9 rejection.

10 Furthermore, the dependent Claims 15-18 and 20-23 of Claim 14 and 19,  
11 respectively, include other limitations that are not taught or suggested by the prior art  
12 of record. Therefore, for at least the above reasons, Applicant respectfully submits  
13 that the §103 rejections of dependent Claims 15-18 and 20-23 is improper, and  
14 respectfully requests reconsideration and withdrawal of this rejection.  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

## Conclusion